

Министерства науки и высшего образования РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, физики, информатики и технологий  
Кафедра информационно-коммуникационных технологий в образовании

# **ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ДЛЯ РАЗРАБОТКИ ВИЗУАЛЬНОГО КОНТЕНТА**

*Выпускная квалификационная работа  
бакалавра по направлению подготовки  
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы ИСИТ-1501  
Института математики, физики, информатики  
и технологий  
Желудкова И.С.

Работа допущена к защите  
« \_\_\_\_ » \_\_\_\_\_ 2019 г.  
Зав. кафедрой \_\_\_\_\_

Руководитель: старший преподаватель,  
Старкова Людмила Николаевна

Екатеринбург – 2019

## Реферат

Желудкова И.С. ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ДЛЯ РАЗРАБОТКИ ВИЗУАЛЬНОГО КОНТЕНТА, выпускная квалификационная работа: 49 стр., рис. 19, табл. 0, библи. 20 назв.

Ключевые слова: ДОПОЛНЕННАЯ РЕАЛЬНОСТЬ, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, ОПЕРАЦИОННАЯ СИСТЕМА ANDROID, НАВИГАЦИЯ.

Объект разработки – программный продукт навигационного назначения с использованием дополненной реальности.

Цель работы – разработка программного продукта, позволяющего использовать технологию дополненной реальности в навигационных целях.

В данной работе представлены и описаны результаты проектирования и разработки программного продукта. Данный продукт использует: устройство на базе android, GPS, спутниковые системы, internet, координаты сети (wi-fi, мобильные сети), доступ к камере устройства. Суть разработанной программы заключается в том, что она накладывает два независимых пространства друг на друга (Окружающий нас мир и виртуальный мир для навигационных целей человека на местности) и позволяет видеть более информативную картину.

Программный продукт реализован на языке Java в среде разработчика android studio.

Программный продукт находится на этапе тестирования и отладки. После данного этапа он сможет быть использован в работе для навигации по зданиям и местности.

## **Оглавление**

<b>ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ .....</b>	<b>4</b>
<b>ВВЕДЕНИЕ.....</b>	<b>5</b>
<b>ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ .....</b>	<b>8</b>
1.1 ТЕХНОЛОГИЯ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ И ЕЕ ИСПОЛЬЗОВАНИЕ .....	8
1.2 ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНСТРУМЕНТА РАЗРАБОТКИ ПРИЛОЖЕНИЙ .....	16
1.3 ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ .....	22
<b>ГЛАВА 2. РАЗРАБОТКИ ВИЗУАЛЬНОГО КОНТЕНТА С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ .....</b>	<b>26</b>
2.1 МОДЕЛЬНЫЕ ПРЕДСТАВЛЕНИЯ ОБЪЕКТА РАЗРАБОТКИ.....	26
2.2 ОПИСАНИЕ ПРОДУКТА (РЕЗУЛЬТАТА РАЗРАБОТКИ) .....	42
2.3 РЕЗУЛЬТАТЫ АПРОБАЦИИ, ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ .....	44
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>47</b>
<b>СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....</b>	<b>48</b>

## **ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

В настоящей работе применяются следующие обозначения и сокращения:

AR – дополненная реальность;

ПО – программное обеспечение;

VR – виртуальная реальность;

ОС – операционная система.

## **Введение**

Дополненная реальность (AR) – технология, позволяющая взаимодействовать человеку с компьютером. Ее специфика заключается в том, что она на программном уровне накладывает два изначально независимых мира: реальный мир и виртуальный мир, созданный человеком и используемый программой.

Вновь стали обсуждать технологию дополненной реальности совсем недавно, а именно 3 года назад (в 2016 году крупные компании начали запускать перспективные проекты виртуальной реальности Vuzix, Sony, ODG, Solos.). Действительно, модифицированная и улучшенная реальность способствует развитию мышления и упрощению понимания многих процессов, (происходит переходе от двумерного мышления к полноценному трехмерному мышлению с понятной визуализацией сложнейших процессов и явлений.

Таким образом технологи AR/VR позволяет перейти от текстового/визуального 2D в 3D-визуальное мышление. Поэтому и развиваются AR-технологии. Дополненная реальность – является перспективнейшей технологией нашего времени, с большим потенциалом развития. Сферы применения данной технологии уже на сегодняшний день выходят за рамки только лишь игровой индустрии: военные технологии, медицина, образование и другие.

Объектом разработки данной работы является программный продукт навигационного назначения с использованием дополненной реальности.

Цель работы – разработка программного продукта, позволяющего использовать технологию дополненной реальности в навигационных целях.

Для достижения поставленной цели необходимо было решить следующие задачи:

1. провести подробный анализ существующих продуктов;
2. выбрать технологию реализации и необходимые программные платформы;
3. разработать программный продукт;

4. подготовить техническую и сопроводительную документацию.

AR формируется благодаря совмещению программных объектов и видео-потока с камеры. Реальный мир приобретает свойство интерактивности за счет взаимодействия со специальными метками – маркерами. Камера устройства выполняет роль глаз системы, а маркеры становятся инструментом воздействия на реальный мир. Камера находит маркеры после чего переносит их в виртуальную среду. Затем происходит наложение одного слоя на другой. Дополненная реальность уже много применялась в той или иной мере в рекламе, медицине, интерактивных играх, управлении и навигации, а также в мониторинге состояния объектов.

Один из примеров, использования AR – маркер карты в интерфейсе камеры на телефоне: чтобы показать, в каком направлении двигаться пользователю. Уже существуют гарнитуры, которые фактически переносят приложение или игру в реальный мир. Более продвинутые AR-системы позволяют обрабатывать стены в доме пользователя, как если бы они были приложениями на экране компьютера.

Многие производители программных и аппаратных продуктов боятся в полной мере инвестировать в столь перспективную технологию. Потенциал ее практически безграничен, но сможет ли она справиться с возложенными на нее надеждами.

Технология, как уже говорилось ранее, в значительной степени упрощает привычные для нас вещи и процессы, за счет этого она и имеет успех в настоящее время. Например, уже существуют приложения по выбору продуктов и услуг, которые позволяют визуализировать продукт, не приобретая его и наряду с удобством возникает экономия времени (время один из самых ценных ресурсов человечества);

Lego и Jurassic World, уже пробуют AR технологии, ответственные программисты и разработчики отмечают, что потенциальное воздействие технологии дополненной реальности на людей колоссально.

Однако, первое такое воздействие на аудиторию произвел именно программный продукт, созданный в развлекательных целях. Этим продуктом стала игра Pokemon Go, которую установило порядка 30 миллионов пользователей по все.

В целях продвижения своей продукции Sephora, Nestlé, Jaguar, Land Rover стали одними из ведущих компаний в области технологий дополненной реальности. Созданные ими программные продукты позволяют визуализировать их продукцию и подробно ознакомиться с характеристиками.

# **Глава 1. ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ**

Данная глава содержит разделы с исторической справкой развития дополненной реальности, а также анализ и рассмотрение готовых продуктов уже разработанных на сегодняшний день. На основе изученных данных, в данной главе будет выбран метод разработки программы, а также составлено подробное техническое задание.

Дополненная реальность – перспективная технология внедрения виртуальных элементов в реальное пространство, с возможностью интерактивного взаимодействия с этими элементами[13]. Одним из основных плюсов данной технологии – получение более информативной картины реального мира при помощи добавления в нее виртуальной информации. То есть дополненная реальность – технология совмещающая виртуальные объекты с окружающим нас миром. На Рисунок 1 показана технология возникновения дополненной реальности с использованием современного компьютера[20].

## **1.1 ТЕХНОЛОГИЯ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ И ЕЕ ИСПОЛЬЗОВАНИЕ**

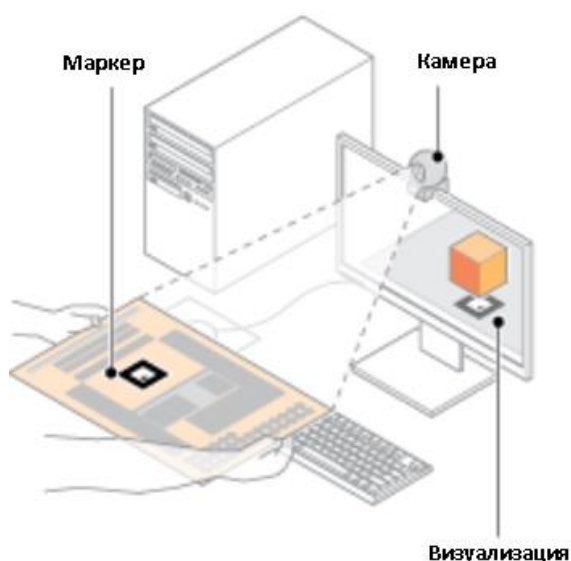


Рисунок 1. Технология создания дополненной реальности



### 1.1.1 Историческая справка

AR технологии начали развиваться еще в 70-ых годах 20-го века. Еще в 1961 году М. Хайлиг представил мультисенсорное устройство, которое напоминало игру с вибрацией и наложенными картинками[12]. Через год, Хайлиг получил патент на виртуальный симулятор "Сенсорама", который был первым в своем роде. Это было очень огромное устройство, похожее на игровые автоматы, которые были популярны в 80-ых годах 20-го века, давало возможность игроку впервые погрузиться в виртуальную реальность: полетать на самолете или прокатиться на машине по любому из городов. Инвесторы посчитали эту идею провальной и дальнейшего развития она не получила, однако, это был первый массовый выход в свет технологии дополненной реальности.

Уже в 1980-ых годах начался следующий этап развития AR. Крюгер разработал программный продукт под названием "Лаборатория искусственной реальности" [14]. Она представляла из себя несколько связанных по сети комнат, в каждой из которых находился большой экран с расположенным позади него видеопроектором. Когда человек заходил в комнату, он видел на экране свое собственное изображение в виде примитивного силуэта, а также подобные силуэты людей в остальных комнатах. У всех "теней" можно было менять цвет или размер, а также присоединять к ним различные визуальные объекты.

Стоит отметить, что термин «дополненная реальность» впервые был введен в 90-ых годах прошлого века. Данный термин родился в стенах компании Boeing Computer Services в Сиэтле.

Все в тех же 90-ых годах военные из США нашли применение этой технологии в своих военно-воздушных силах. Их программа позволила разработать методику обучения пилотов. Помимо картинок военные применяли полное погружение в процесс обучения, дополнив реальность звуком и моделями кабин самолетов с реальными приборами управления.

Через пару лет после военных уже малые компании начали выпускать инструкции с дополненной реальностью для своих продуктов (принтеры, пылесосы и т.д.).

Однако старое двумерное мышление не давала ученым принять и понять потенциал дополненной реальности до конца 2000-ых годов, а также затрудняла ее массовое использование тот факт, что при работе технологии требовались сложные программные решения и гигантское по габаритам оборудование.

Программный продукт ARToolKit стал очередным прорывом в области AR. После разработки в Японии в 2000-ых годах, оно позволило на портативных устройствах выделять кадры из видеопотока и накладывать виртуальные элементы в 3D. ARToolKit дал понять, что эта технология может быть доступна каждому человеку, ее мобильность более не ограничивалась размерами аппаратной части специально разработанных устройств.

В начале 21 века была разработана очередная игра – ARQuake, которая стала прообразом шлемов дополненной и виртуальной реальности. Именно данный продукт вызвал новую волну интереса к данной технологии, выявил основные направления ее развития: миниатюризация аппаратной части и разработка новых, более оптимальных алгоритмов работы.

Уже через 8 лет (2009 год) AR технология начала активно внедряться в виде программного обеспечения в смартфоны. Этот год стал началом общедоступности, ранее недоступной широкому кругу людей, технологии.

Данные приложения практически сразу стали доступны на всех существующих на тот момент платформах (Android, IOS, Windows mobile и др.). Можно с уверенностью сказать, что технология AR находилась в искусственном сне, до момента появления подходящей аппаратной части, сочетающей в себе камеру и вычислительные мощности необходимые для обработки сложнейших алгоритмов.

В 2016 году была запущена игра «Pokemon Go», в которой нужно ходить по реальным объектам и находить виртуальных монстров. Игра стала самым массовым применением AR (число скачиваний более 20 миллионов).

В настоящее время инвестиции в разработку программ и оборудования для виртуальной реальности составляют порядка 2 миллиардов долларов.

### **1.1.2 Применение дополненной реальности**

Технология дополненной реальности в образовании применяется для реконструкции исторических событий в режиме реального времени. Стоит отметить что данная технология не только позволяет понять абстрактные процессы обучающимся, но и помогает учителям организовывать сбалансированный и интересный учебный процесс. Например, в Японии уже используются приложения, которые работают на простых смартфонах и позволяют в учебниках видеть анимированные процессы, персонажей на тех страницах где это необходимо.

На сегодняшний день второй по важности отраслью, осваиваемой данной технологией, является сфера здравоохранение. Врачи учатся на протяжении всей жизни и без должной практики стать хорошим врачом невозможно. Дополненная реальность позволяет не только обучать врачей, но и не дает совершать критические ошибки в ходе сложных операций.

Военные тоже интересуются технологией. Американская компания BAE Systems разработала шлем под названием Striker II, в котором вместо очков используется своеобразный козырек с дисплеем. На него проецируется изображение с камеры ночного видения, а аппарат способен отслеживать движения головы оператора. Так данные всегда располагаются в направлении взгляда пользователя.

Еще одна американская компания Matterport с помощью AR создает виртуальный рынок продажи недвижимости.

Но, естественно, главные драйверы AR, как и многих других технологий, - гиганты Apple, Google и Microsoft. Они активно инвестируют средства в AR, чтобы сделать технологию более эффективной и доступной для миллиардов пользователей смартфонов.

Дополненная реальность – одна из многих технологий взаимодействия человека и компьютера[15]. Совмещение реальностей может найти применение не только в перечисленных сферах. Перспективы данной технологии могут затронуть практически все стороны человеческой жизни, так как мы живем в век информационных технологий, а AR дает возможность получить эту информацию в полной мере.

Стоит отметить и продвижение приборостроения. Ведь без миниатюризации персональных устройств и роста их вычислительной мощности ни дополненная реальность, ни виртуальная реальность вряд ли бы имели место, и оставались бы в качестве недостигаемой мечты.

### **1.1.3 Обзор существующих программных продуктов**

Дополненная реальность — это интеграция виртуальных объектов (как правило, трехмерных, но не обязательно) в реальный мир. Возможных форматов реализации — множество: начиная от проектов наподобие Google Glass (так, впрочем, не получивших массового распространения) и навигационного оборудования (мы писали о проекте AR-мотошлема) и заканчивая приложениями, использующими камеру смартфона или планшета.

Применение дополненной реальности растет бешеными темпами. Это вызвано развитием технологии искусственного интеллекта и их совместным использованием. Тут стоит вновь отметить важность роста производительности устройств и уменьшение их габаритов.

Google, Apple и Microsoft – начали проявлять огромное внимание именно дополненной реальности. Эти компании имеют огромную техническую базу (способны с 0 разработать аппаратную часть и написать уникальную программу

под него), а также практически неограниченный бюджет на инновации. Лидерство этой тройки в настоящее время не вызывает сомнений.

В 2012 году Google Glass были представлены корпорацией Google, как инструмент для разработчиков. Аппаратная часть данного устройства миниатюризирована и представляет собой обычные очки со встроенной камерой и дисплеем. Однако начало работы этой корпорации не обвенчалось успехом, ведь цена данного устройства была высока, и оно не стало пользоваться популярностью среди широкого круга пользователей. Однако, работы над данным девайсом ведутся и по сей день.

Одной из причин провала устройства «Гугл очков» помимо цены являлись бесконечные скандалы с конфиденциальностью данных. Однако Google имеет не менее амбициозные проекты в области дополненной реальности, такие как: позиционирование и навигация в пространстве, продвижение и реклама продукции и многое другое.

А, например, шведская фирма Ikea создала свой AR продукт – приложение, с помощью которого позволяет пользователям проверить, как мебель может выглядеть в их домах. Данная программа присутствует на всех мобильных платформах. Пример работы данной программы представлена на Рисунок 2. Данный вариант программы позволяет пользователям с использованием своего смартфона на базе IOS или android визуально оценить мебель и подобрать ее под свой интерьер, не покидая своего дома.

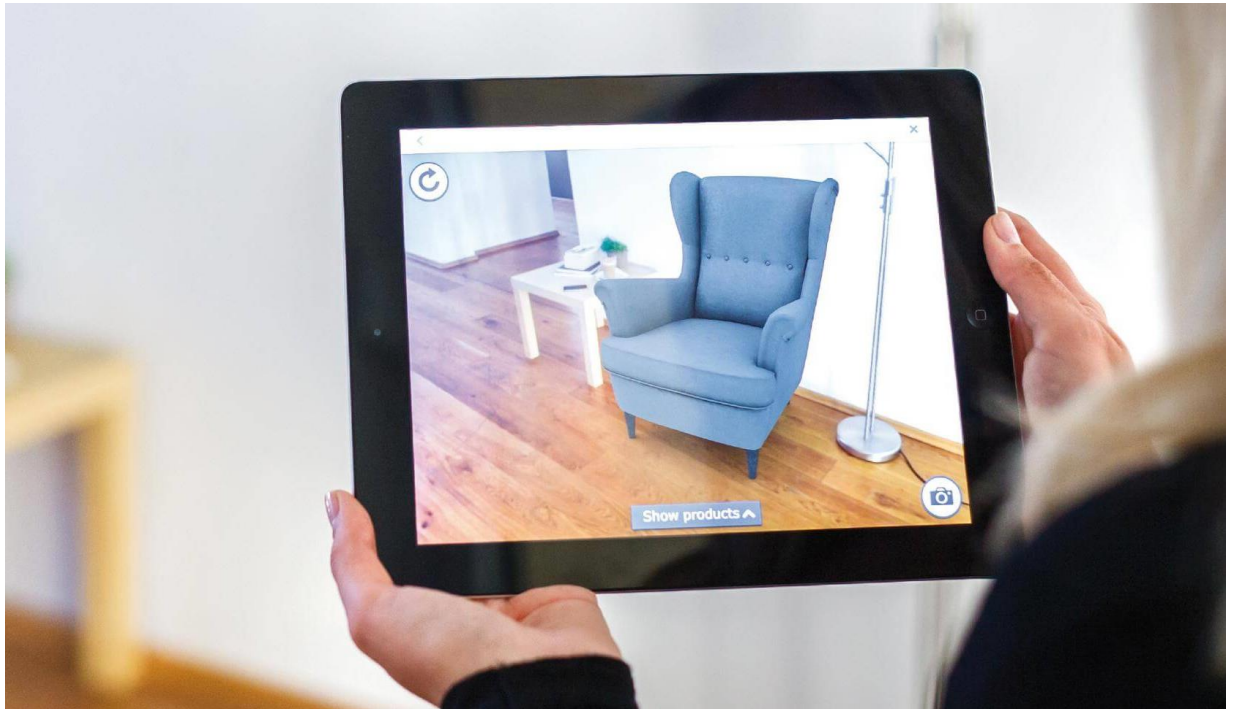


Рисунок 2. Пример работы программы IKEA place

Заслуги Apple в отрасли AR пока скромны, однако компания продолжает "держат руку на пульсе". Главное достижение компании – технология дополненной реальности ARKit, которая позволяет распознавать габариты реальных объектов и учитывать условия освещения.

Данный продукт максимально достоверно внедряет виртуальные объекты в реальную жизнь. Совместимая с iOS технология может стать самой массовой платформой дополненной реальности в мире, учитывая интерес к продуктам компании.

Пример работы программы ARKit представлен на Рисунок 3. Данный программный продукт универсален. Применение может быть найдено как в развлекательных целях, так и навигации продвижении товаров и услуг. Так же данный продукт AR уже применяется в образовании.



Рисунок 3. Пример работы программы ARKit

Microsoft также изо всех сил стремится к созданию дополненной реальности в качестве платформы для будущего мобильных технологий. В компании объединили виртуальные и дополненные миры для пользователей, создав, возможно, первое представление о смешанной реальности благодаря гарнитуре Microsoft HoloLens. От других устройств дополненной реальности они отличаются тем, что переняли у VR-шлемов возможность отслеживать мельчайшие движения головы пользователя. Процесс происходит при помощи обычного гироскопа и акселерометра и позволяет не только ускорить обработку данных в особых случаях, но и дополнить управление жестами.

Кроме того, Microsoft уже несколько лет продолжает разрабатывать платформу смешанной реальности Windows Mixed Reality, заявленную как часть операционной системы Windows 10. По данным разработчиков, она обеспечивает "голографический опыт" в смешанной реальности с совместимыми шлемами. Вероятно, в компании рассчитывают, что с развитием AR пользователям потребуется унифицирующий интерфейс и общая платформа.

### **1.1.4 Оборудование для технологии дополненной реальности**

Для работы с технологией дополненной реальности обязательно необходимы следующие компоненты:

- 1) Графическая станция. Это может быть мобильный телефон, ноутбук, персональный компьютер, графическая рабочая станция с профессиональной видеокартой. Одним словом, компьютер.
- 2) Дисплей. Экран телефона, телевизор, монитор, моно или стерео дисплей, проекционный экран.
- 3) Камера. Благодаря камере мы получаем «слепок» реального мира, на который специальное программное обеспечение накладывает виртуальные объекты.
- 4) Метки, или маркеры.

## **1.2 ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНСТРУМЕНТА РАЗРАБОТКИ ПРИЛОЖЕНИЙ**

Математические алгоритмы, которые позволяют камере увидеть и распознать метку (маркер) в окружающем пространстве, а затем определить, что за объект привязан к текущему маркеру, и в итоге наложить эту модель на метку таким образом, чтобы эта модель повторяла Движение физической метки.

Технология дополненной реальности – это программное обеспечение, которое с помощью определенных алгоритмов связывают данные камеры, маркеры и компьютер в единую сеть, с которой пользователь может взаимодействовать в той или иной мере.

У программы AR стоят следующие задачи:



- 1) определить трехмерное положение реальной метки по ее снимку, полученному с помощью камеры;
- 2) Распознать эту метку.
  - a) Сначала снимается изображение с камеры.
  - b) Затем программа распознает пятна на каждом кадре видео в поисках заданного шаблона – рамки метки.
  - c) Как только камера «находит» в окружающем пространстве рамку, ее следующая задача – определить, что именно изображено внутри рамки.
  - d) Последней задачей становится наложение объекта, соответствующего метке.
- 3) Сохранение согласованности метки и наложенного на нее объекта.

При перемещении камеры относительно объекта будет рассчитано смещение виртуальной метки в режиме реального времени.

### **1.2.1 Обоснование постановки задачи**

Разработка программного продукта, позволяющего использовать технологию дополненной реальности в навигационных целях является очень важной задачей.

Хотя программные продукты подобного типа уже существуют, они не подходят для навигации по зданию университета, так как разрабатывались они с целью более масштабных перемещений и «маркеры» в рамках одного здания подробно не выставлялись. А заказ программы для навигации по университету или любому другому зданию у ведущих разработчиков (google, samsung, apple, sony и др.) потребует вложения значительных средств.

Существуют три основных подхода программирования для достижения поставленной цели:

1. «Безмаркерная» технология. Работает на подобии карты на которой заранее нанесены опорные точки. Создание карты происходит на подобии рисования

сетки, узлы которой и являются опорными точкам. Простота данной технологии в основном проявляется в ненужности создания меток и их описаний. Однако при данной модели необходимо использовать расчёт координат от узлов созданной виртуальной карты.

2. Второе направление – технолог AR на базе специальных маркеров, или меток. Является наиболее распространённой на сегодняшний день из-за ее стабильности и надёжности. Данный подход требует меньшей вычислительной мощности, при условии, что метка будет описана достаточно точно и камера сможет узнать ее. Стоит отметить что данный подход зачастую используется в комбинации с другими методами.
3. «Пространственная технология». Данные о расположении искомой метки для камеры исходят из пространственного положения реального объекта на местности относительно пользователя. Данный подход использует данные GPS/ГЛОНАСС, гироскопа и компаса, встроенных в мобильный телефон. Место виртуального объекта определяется координатами в пространстве. Активация программы дополненной реальности происходит при совпадении координаты, заложенной в программе, с координатами пользователя[16].

Так же, использование маркерной технологии имеет дополнительные преимущества в плане внедрения в методическую часть наглядных печатных материалов, используемых в общеобразовательных учреждениях при изучении конкретной темы и проведении практических работ по ней.

Стоит отметить, что при применении маркерной технологии, можно столкнуться с некоторыми проблемами. Одной из основных проблем является потеря метки. Камера перестает видеть объект и за счет этого происходит нарушение работоспособности всей программы.

Суть данной проблемы может быть описана несколькими причинами:

- 1) Освещение. То есть если в кадре присутствует затемнение или же наоборот слишком яркое освещение. Светочувствительность камеры может не позволить распознать метку.
- 2) Вторая проблема – Что камера не всегда полностью покрывает объект и поэтому происходит исчезание метки. Однако данная проблема решается путем комбинирования алгоритмов пространственного и маркерного.
- 3) Низкий FPS камер не позволяет в реальном времени быстро отследить изменение положение и за счет этого программа не может сфокусироваться на метке.
- 4) Более серьезная проблема – это калибровка камеры. Калибровка применяется для того чтобы модель реальной камеры могла быть воссоздана в компьютерном пространстве. (Для того чтобы добавить перспективу и глубину в 2D картинку, которая отображается с камеры на экран, нужно определить параметры перспективной проекции для камеры. Это можно сделать в домашних условиях, используя «шахматную доску» и специальное программное обеспечение).
- 5) Низкая разрешающая способность камеры. Потеря метки наиболее вероятна если камера практически не видит исходной метки. Однако покупка более качественного фото/видео оборудования позволяет с легкостью избежать данной неприятности.
- 6) И последняя проблема – это программное обеспечение. Здесь основополагающую роль имеет тот факт, что не существует идеальных алгоритмов работы, и программист должен сам определять: использовать более надежный, но менее быстрый алгоритм или же наоборот в угоду быстродействию быть готовым столкнуться с рядом проблем.

Аппаратная часть, для реализации базовых функций технологии дополненной реальности должна решать 3 основных задачи: получать видеопоток хорошего качества, иметь возможность обработать данный видеопоток и дополнить слоем с виртуальными объектами и, конечно же, вывести обработанные данные на устройства вывода для восприятия конечным пользователем[17]. В качестве аппаратной части было выбрано устройство на базе android 8.0.

### 1.2.2 Выбор метода разработки

Разработка велась в интегрированной среде разработчика Android Studio на языке Java по инкрементной модели которая схематически представлена на Рисунок 4.



Рисунок 4. Инкрементный метод программирования

В инкрементной модели полные требования готовому продукту вытекает из требований к различным сборкам, применяемым в программе. Программирование происходит поэтапно, что означает что при выявлении неработоспособности одного из блоков не нужно будет переписывать весь код, достаточно работать только с тем самым блоком.

Так же в данной модели имеют место несколько различных циклов разработки. Цикл делится на еще меньшие модули. К каждой с такой структурной единице можно писать собственные требования, что облегчает процесс программирования.

Процедура разработки по инкрементной модели предполагает выпуск на первом большом этапе продукта с минимальным функционалом (базовая функциональность), а затем уже постепенное добавление новых функций. Процесс продолжается до тех пор, пока не будет создана полная система.

Инкрементная модель применима там, где отдельные запросы на программный продукт четко понятны и поэтому они могут быть легко реализованы[18].

Основное преимущество для нас в данной модели заключается в том, что некоторые детали могут дорабатываться с течением времени и в целом программа не потеряет своей работоспособности полностью.

Стараясь исключить технологические риски и обойти проблемные моменты, при разработке прототипа программного комплекса, мы остановили свой выбор на надежной и проверенной маркерной технологии дополненной реальности совместно с использованием пространственной технологии (определение позиции с помощью GPS и мобильных данных).

### **1.2.3 Инструмент разработки**

Как говорилось в предыдущем разделе в качестве инструментария выступает интегрированная среда разработки Android Studio.

Данная интегрированная среда разработки находится в свободном доступе начиная с 2013 года. На Рисунок 5 представлен интерфейс программы, выбранной в качестве среды разработки.

Помимо бесплатности, основным преимуществом данного инструмента разработчика выступает его широкая функциональность и поддержка многих

языков программирования. Так же данный программный продукт разработан специально для программирования для устройств на базе Android.

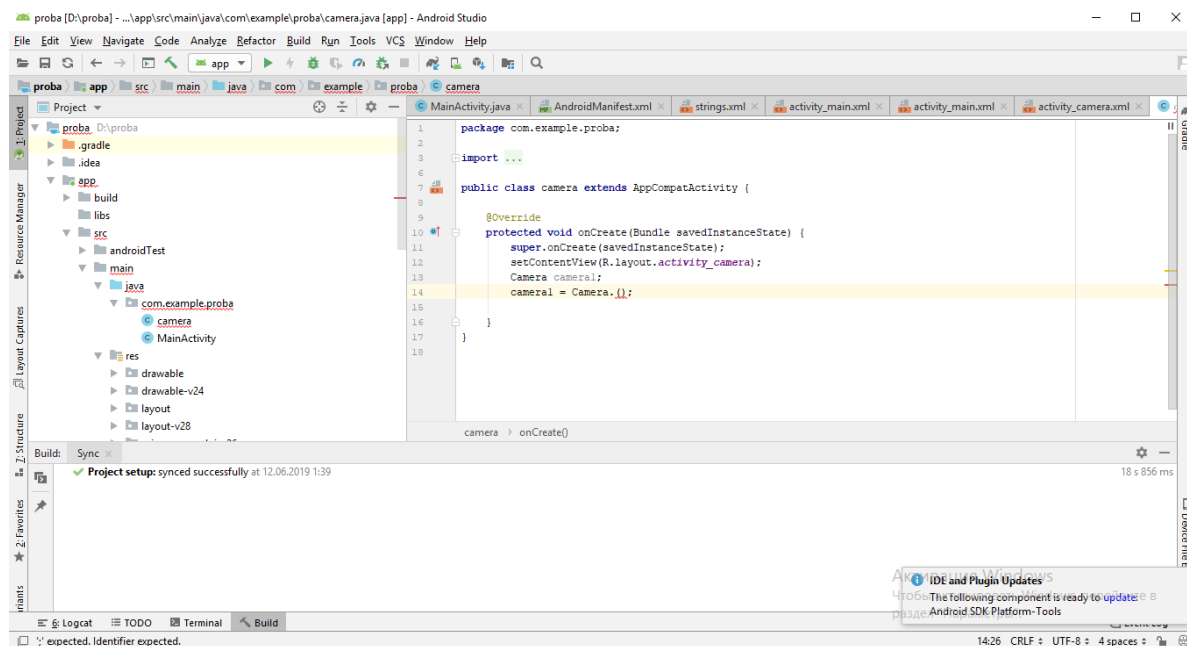


Рисунок 5. Интерфейс Android Studio

Android Studio, в ходе программирования, позволяет разработчику наглядно видеть любые изменения, которые производились в реальном времени с приложением[19]. Так же функционал программы позволяет визуализировать работу программы на устройствах с различными разрешениями экрана (различных типах смартфонов и планшетов на базе андроид) что является ощутимым плюсом, так как на сегодняшний день насчитывается более сотни крупных производителей электронных девайсов и смартфонов.

### 1.3 Формализованное описание технического задания

Для реализации программного продукта с учетом имеющихся технических средств было составлено техническое задание.

Техническое задание

#### 1. Общие сведения

##### 1.1. Название организации заказчика.

ФГБОУ ВО «Уральский государственный педагогический университет»,  
Институт математики, физики, информатики и технологий, кафедра  
информационно-коммуникационных технологий.

1.2. Название продукта разработки (проектирования).

Мобильное приложение с дополненной реальностью «Путеводитель  
УрГПУ»

1.3. Назначение продукта.

Проинформировать потребителей об нахождении корпусов, кабинетов  
УрГПУ, убедить пользователя в том, что именно с данным мобильным  
приложением он не потеряется в УрГПУ.

1.4. Целевая аудитория.

Студенты ВУЗа, абитуриенты, посетители (гости и участников различных  
мероприятий)

1.5. Плановые сроки выполнения

Начало работ: 01.09.2018. Окончание работ: 20.05.2019.

2. Характеристика области применения.

2.1. Процессы и структуры, в которых предполагается использование  
продукта разработки.

Информационным студенческим мобильным приложением  
«Путеводитель УрГПУ».

2.2. Характеристика пользователей

2.2.1. Разработчик

Должен иметь навыки работы с языками программирования (Java),  
интегрированной средой разработки для работы с платформой Android.

2.2.2. Конечный пользователь

Должен иметь навыки работы со смартфоном, уметь устанавливать  
приложения на смартфон.

3. Требования к продукту разработки

3.1. Требования к продукту в общем

Мобильное приложение состоит из 3 окон интерфейса:

- Выбор корпуса
- Выбор кабинета/аудитории
- Камера с гидом

В интерфейсе «Выбор корпуса» должны быть кнопки выбора с адресами корпусов.

В интерфейсе «Выбор кабинета/аудитории» должны быть кнопки выбора аудиторий.

В интерфейсе «Камера с гидом» должна открываться камера, также на видеопоток необходимо наложить гида (стрелки) и метку пункта назначения (если она входит в угол камеры, иначе показать стрелку куда поворачивать камеру). Необходимо вывести на экран выбор пункта назначения и сколько метров осталось до назначенной цели.

Пунктами назначения являются учебные корпуса, также кабинеты (ректор, декан, бухгалтерия, профсоюз и т.п.).

### 3.2.Аппаратные требования

#### 3.2.1. При разработке

3.2.1.1. Персональный компьютер.

3.2.1.2. Процессор: Intel 2,4 МГц.

3.2.1.3. ОЗУ: 8 Гб.

3.2.1.4. Дисковое пространство (для установки Android Studio): 8 Гб.

#### 3.2.2. .При эксплуатации

3.2.2.1. Смартфон.

3.2.2.2. Android 8.0

### 3.3.Программное обеспечение

#### 3.3.1. 5.1.При разработке

3.3.1.1. JDK 12 или выше;

#### 3.3.2. При эксплуатации

Android 8.0 и выше.



### 3.3.3. Порядок взаимодействия с другими системами

Не предусмотрено

## 4. Требования к пользовательскому интерфейсу

### 4.1.Общая характеристика пользовательского интерфейса.

Интерфейс программы представляет из себя интерфейс видеокамеры с дополнительным меню выбора корпуса/кабинета/аудитории на котором при наведении камеры с помощью маркеров будут указываться направление движения и подпись выбранного кабинета/аудитории. С помощью геолокации будет рассчитано расстояние до искомого объекта.

## 5. Требования к документированию

### 5.1.Перечень сопроводительной документации.

Техническое задание.

## 6. Порядок сдачи приема продукта

В соответствии со сроками сдачи выпускной квалификационной работы: с 1.09.2019 по 21.05.2019.

## **Глава 2. РАЗРАБОТКИ ВИЗУАЛЬНОГО КОНТЕНТА С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ**

### **2.1 Модельные представления объекта разработки**

Структура приложения дополненной реальности, которое служит для навигации и визуализации направления до метки, состоит из пяти основных модулей:

1. Модуль отслеживания камеры;
2. Модуль хранения объектов;
3. Модуль визуализации;
4. Модуль геолокации;
5. Модуль пользовательского интерфейса.

Кадры с камеры передается в модуль отслеживания камеры, которые после обработки по определенным алгоритмам совместно с модулем геолокации обрабатывают видеопоток: происходит поиск метки на местности и расчёт положения камеры относительно метки с помощью азимутального угла и углов места. Когда все углы определены и точно известно положение камеры относительно метки, из модуля хранения объектов на видеопоток размещается необходимый для визуализации объект. Визуализации происходит в реальном времени. Структура приложения представлена на Рисунок 6.

Алгоритм работы программы представлен на Рисунок 7. Алгоритм работает по следующему сценарию:

- 1) Происходит инициализация камеры.
- 2) Получение видеопотока инициализированной камеры в реальном времени.
- 3) Выделение кадров с последующей обработкой.
  - а) Поиск метки на выделенном кадре.

b) В случае, когда метка попала в поле зрения камеры, происходит ее идентификация.

c) Метка не попала в кадр – приложение выделяет новый кадр также если маркер не удалось идентифицировать, то приложение также возвращается к шагу получения кадра из видеопотока.

d) В случае успешной идентификации метки запускается алгоритм преобразования виртуальных объектов (стрелок и отметки цели) и позиционирование данного объекта.

e) Визуализация виртуальных объектов поверх видеопотока.

f) Последним этапом является опрос пользовательского ввода.

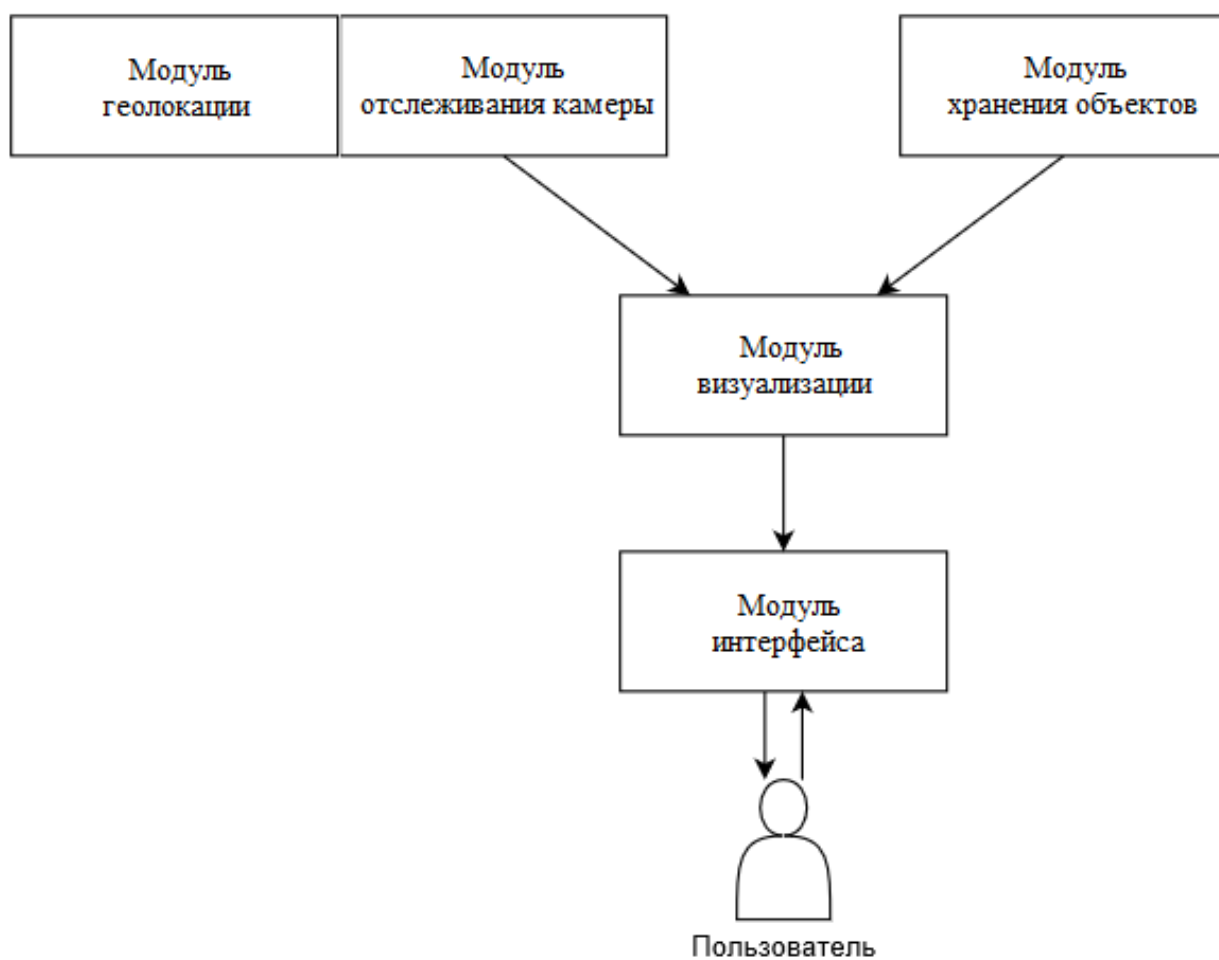


Рисунок 6. Структура приложения дополненной реальности навигационного назначения

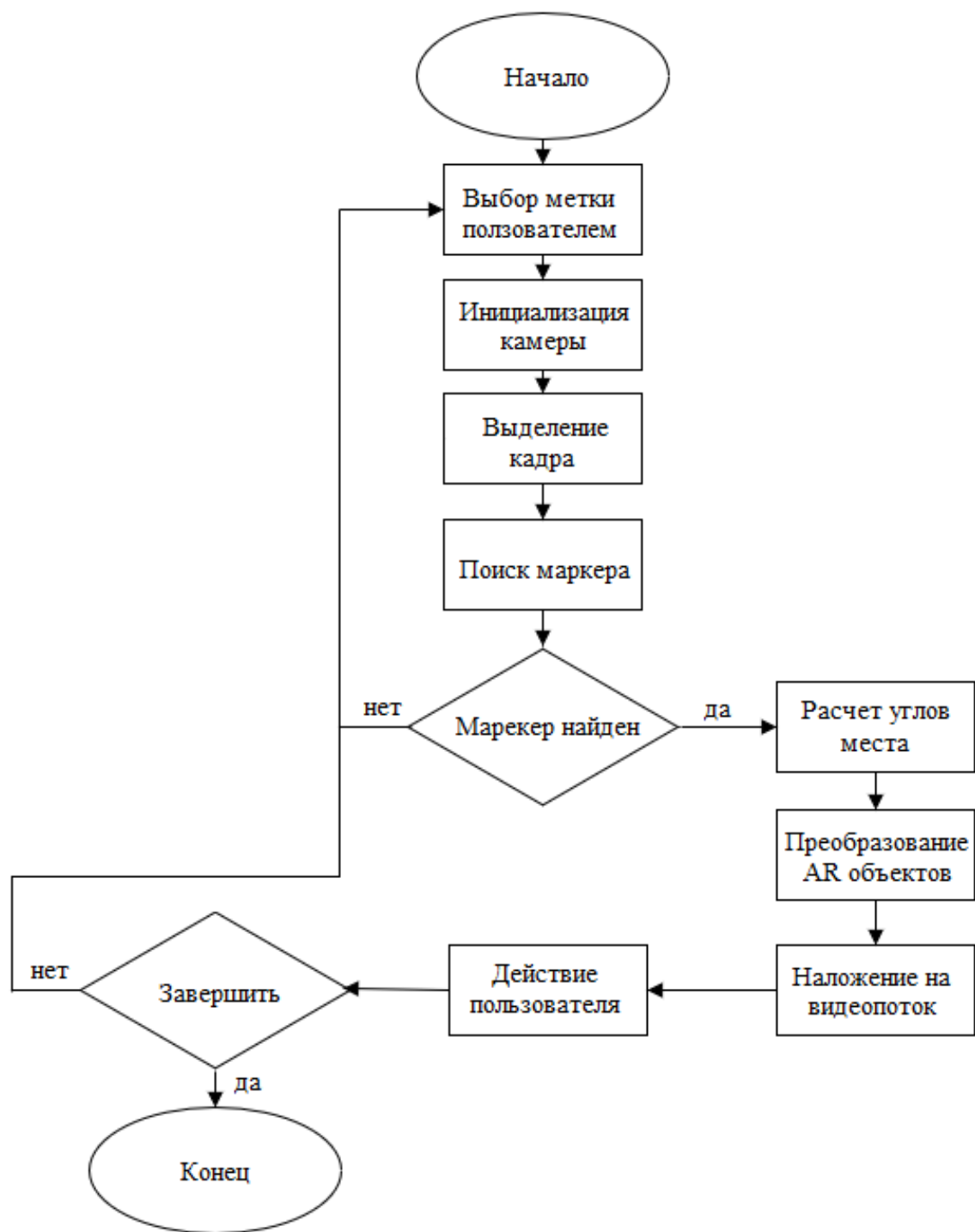


Рисунок 7. Алгоритм работы программы

Перейдем к рассмотрению этапов программирования. Сначала рассмотрим, как был создан интерфейс программы «Путеводитель УрГПУ» – меню с выбором списка корпусов. Для создания кнопок на интерфейсе программы был использован следующий код:

```
<Button
```

```
android:id="@+id/my_KL9"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="@string/my_Button_text"/>
```

Аналогично создается кнопка для других корпусов. Интерфейс с добавленными кнопками представлен на

Рисунок 8. Аналогично создавалось следующее меню с выбором места назначения, которое представлено на Рисунок 9.

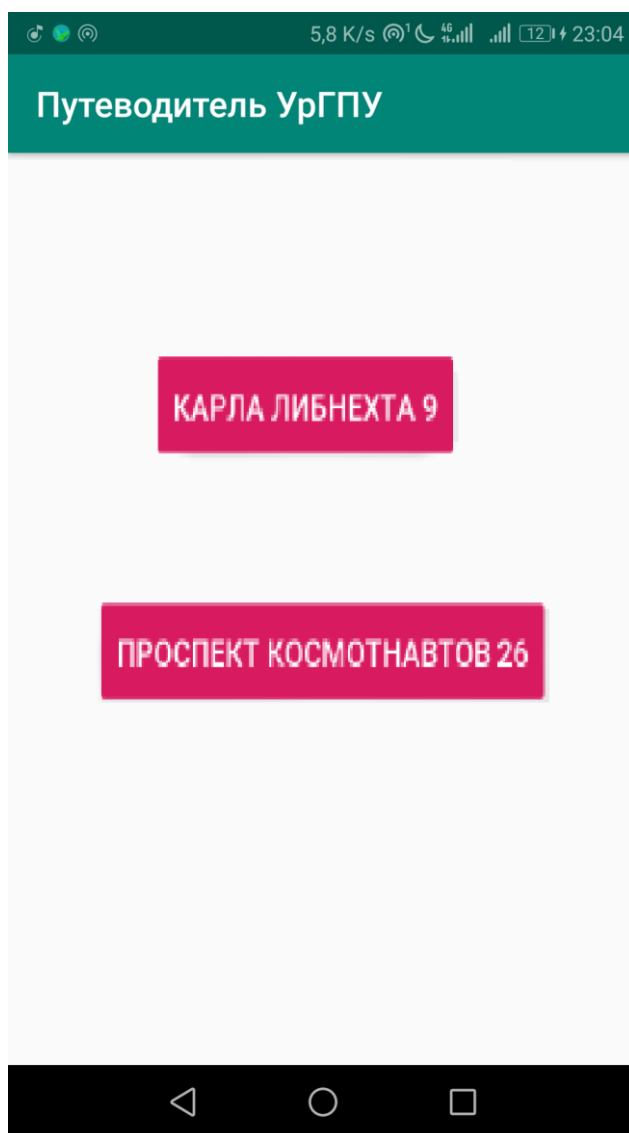


Рисунок 8. Интерфейс программы «Путеводитель УрГПУ»

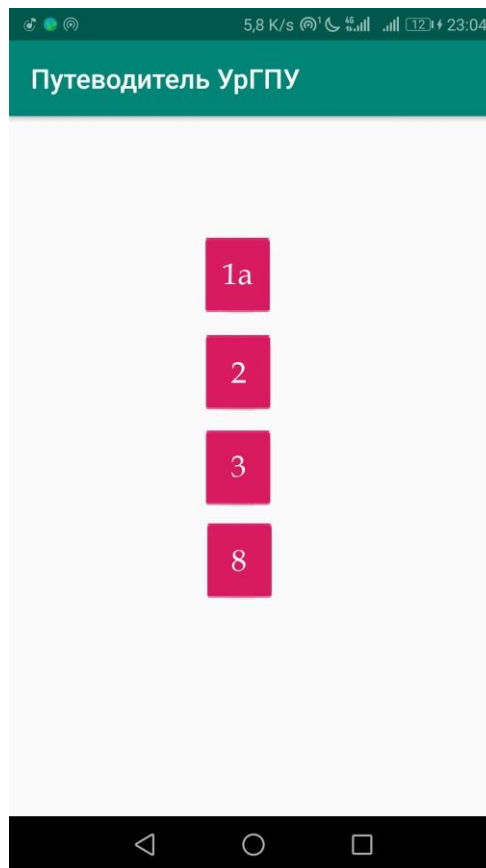


Рисунок 9. Интерфейс программы «Путеводитель УрГПУ»

Следующим этапом разработки программы стал добавление кода, позволяющего получить свои координаты через Maps API.

Для того чтобы использовать новый Maps API, мы предоставили Google имя пакета нашего приложения. Затем в проекте, откроем файл манифеста `AndroidManifest.xml`. Просто перед тегом `</application>` необходимо объявить две переменных данных. Первый тег работает с библиотекой Google Play Services, добавленной в приложение. Второй содержит наш API ключ, полученный из консоли. Так же нам нужно указать разрешения для нашего приложения. Итогом данных действий стал следующий код:

```
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="Maps API" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

```
<uses-permission  
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Для получения доступа к камере устройства был прописан следующий код:

```
<SurfaceView  
    android:id="@+id/cameraview"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content" />
```

После выбора учебного корпуса в результате выполнения данного кода откроется камера. Пример скриншота приведен на Рисунок 10.

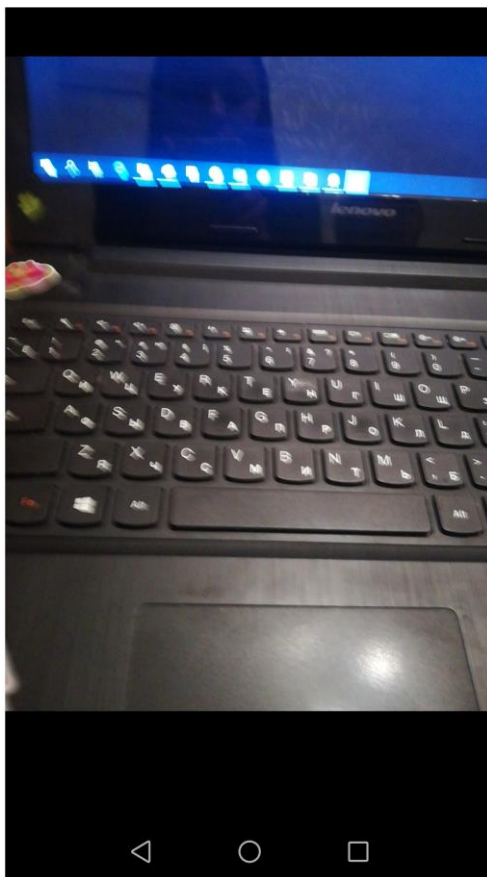


Рисунок 10. Интерфейс программы «Путеводитель УрГПУ»

Нам понадобятся два интерфейса первый из которых будет является слушателем изменения местоположения, а второй слушателем изменения азимута и углов места. Код создания слушателей представлен ниже:



```
import android.location.Location;

public interface OnLocationChangeListener {
    void onLocationChanged(Location currentLocation);
}
public interface OnAzimuthChangeListener {
    void onAzimuthChanged( float azimuthFrom, float azimuthTo);
}
```

Далее решали задачу навигации от текущей точки до одного из маркеров с использованием азимутального угла. Как видно из кода, приведенного ниже, здесь используется переменная — виртуальный датчик, TYPE\_ROTATION\_VECTOR. Входной информацией для работы данной переменной является выходная информация с датчиков устройства (смартфона) а именно: гироскопа, акселерометра с датчика магнитного поля. И если акселерометр есть в каждом устройстве, то датчиком магнитного поля (в старых бюджетных телефонах последний зачастую отсутствует). Следовательно, старых устройствах углы определяться не будут, отсюда вытекает одно из ограничений использования данной программы на устаревших смартфонах.

```
import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;

public class MyCurrentAzimuth implements SensorEventListener {

    private SensorManager sensorManager;
    private Sensor sensor;
    private int azimuthFrom = 0 ;
    private int azimuthTo = 0 ;
    private OnAzimuthChangeListener mAzimuthListener;
    Context mContext ;
    //в конструкторе передаем интерфейс OnAzimuthChangeListener и контекст
    public MyCurrentAzimuth(OnAzimuthChangeListener azimuthListener, Context context) {
        mAzimuthListener = azimuthListener;
        mContext = context;
    }
    //подключаемся к сенсору и регистрируем слушатель для данного датчика с заданной
    периодичностью
    //SENSOR_DELAY_UI - частота обновления пользовательского интерфейса.
```

```

//TYPE_ROTATION_VECTOR - Возвращает положение устройства в пространстве в виде
угла
//относительно оси Z, указывающей на север.
// Виртуальный датчик, берущий показания от акселерометра, гироскопа и датчика
магнитного поля.

    public void start(){
        sensorManager = (SensorManager) mContext .getSystemService( mContext.
SENSOR_SERVICE);
        sensor = sensorManager .getDefaultSensor(Sensor. TYPE_ROTATION_VECTOR);
        sensorManager.registerListener( this, sensor ,
            SensorManager. SENSOR_DELAY_UI);
    }
//Отменяет регистрацию слушателя для всех датчиков.
    public void stop(){
        sensorManager.unregisterListener( this );
    }

//вызывается при новом событии датчика
//получаем матрицу вращения устройства
// в переменную azimuthTo сохраняем градусную меру угла поворота в радианах
@Override
    public void onSensorChanged(SensorEvent event) {
        azimuthFrom = azimuthTo;

        float[] orientation = new float[ 3];
        float[] rMat = new float[ 9];
        SensorManager.getRotationMatrixFromVector( rMat, event. values);
        azimuthTo = ( int) ( Math. toDegrees( SensorManager.getOrientation( rMat, orientation ) [ 0] )
+ 360 ) % 360 ;

        mAzimuthListener.onAzimuthChanged( azimuthFrom , azimuthTo );
    }

@Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }
}

```

Далее был написан код для выставления меток, он имеет следующий вид:

```

public class KL9{
    private String mName;
    private double mLatitude ;
    private double mLongitude ;

    public KL9 (String newName,
                double newLatitude, double newLongitude) {
        this. mName = newName;
    }
}

```

```

        this.mLatitude = newLatitude;
        this.mLongitude = newLongitude;
    }

    public String get KL9Name() {
        return mName;
    }
    public double get KL9Latitude() {
        return mLatitude;
    }
    public double get KL9Longitude() {
        return mLongitude;
    }
}

```

После того как подготовили данные от датчиков, пришло время для реализации главного класса `CameraViewActivity`. Первая и наиболее важная вещь заключается в реализации интерфейса `SurfaceHolder.Callback` который позволяет отправить изображение с камеры в макет и обработать связанные с этим различные события. Интерфейс реализует три метода, ответственных за это: `surfaceChanged()` `surfaceCreated()` `surfaceDestroyed()`.

Класс `CameraViewActivity` имплементирует также интерфейсы `OnLocationChangeListener`, `OnAzimuthChangeListener`, с их методами `onLocationChanged` и `onAzimuthChanged`, а также интерфейс `View.OnClickListener` — слушатель нажатия кнопки с методом `onClick`. Код представлен ниже:

```

import android.app.Activity;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.graphics.PixelFormat;
import android.hardware.Camera;
import android.location.Location;
import android.os.Bundle;
import android.view.SurfaceHolder;
import android.view.SurfaceView;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import java.io.IOException;
import java.util.ArrayList;

```

```

import java.util.List;

public class CameraViewActivity extends Activity implements
    SurfaceHolder.Callback, OnLocationChangeListener, OnAzimuthChangeListener,
    View.OnClickListener {
    //объявляем необходимые переменные
    private Camera mCamera;
    private SurfaceHolder mSurfaceHolder;
    private boolean isCameraviewOn = false ;
    private KL9 mKL9;

    private double mAzimuthReal = 0 ;
    private double mAzimuthTeoretical = 0 ;

    /*нам понадобятся, помимо прочего, две константы для хранения допустимых откло-
    нений дистанции и азимута устройства от целевых. Значения подобраны практически, вы
    можете их менять, чтобы облегчить, или наоборот, усложнить задачу поиска метки. Точ-
    ность дистанции указана в условных единицах, равных примерно 0.9м, а точность азимута - в
    градусах*/

    private static final double DISTANCE_ACCURACY = 20 ;
    private static final double AZIMUTH_ACCURACY = 10 ;

    private double mMyLatitude = 0 ;
    private double mMyLongitude = 0 ;

    public static final double TARGET_LATITUDE = 27.590377 ;
    public static final double TARGET_LONGITUDE = 14.425153 ;

    private MyCurrentAzimuth myCurrentAzimuth;
    private MyCurrentLocation myCurrentLocation;

    TextView descriptionTextView;
    ImageView pointerIcon;
    Button btnMap;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_camera_view);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        setupListeners();
        setupLayout();
        setAugmentedRealityPoint();
    }
    //создаем экземпляр метки с указанием координат его местоположения

    private void setAugmentedRealityPoint() {

```

```

        mKL9 = new KL9(
            getString(R.string.p_name ),
            TARGET_LATITUDE, TARGET_LONGITUDE
        );
    }
    /*вычисляем дистанцию между устройством и меткой по формуле. Результат прихо-
    дит в десятичных градусах, умножение его на 100000 дает некую условную единицу, при-
    близительно равную 0.9м.*/

    public double calculateDistance() {
        double dX = mKL9.getLatitude() - mMyLatitude;
        double dY = mKL9.getLongitude() - mMyLongitude;

        double distance = (Math.sqrt(Math.pow(dX, 2) + Math.pow(dY, 2))) * 100000;

        return distance;
    }
    /*вычисляем теоретический азимут по формуле. Вычисление азимута для разных чет-
    вертей производим на основе таблицы. */

    public double calculateTeoreticalAzimuth() {
        double dX = mKL9.getLatitude() - mMyLatitude;
        double dY = mKL9.getLongitude() - mMyLongitude;

        double phiAngle;
        double tanPhi;
        double azimuth = 0;

        tanPhi = Math.abs(dY / dX);
        phiAngle = Math.atan(tanPhi);
        phiAngle = Math.toDegrees(phiAngle);

        if (dX > 0 && dY > 0) { // I quarter
            return azimuth = phiAngle;
        } else if (dX < 0 && dY > 0) { // II
            return azimuth = 180 - phiAngle;
        } else if (dX < 0 && dY < 0) { // III
            return azimuth = 180 + phiAngle;
        } else if (dX > 0 && dY < 0) { // IV
            return azimuth = 360 - phiAngle;
        }

        return phiAngle;
    }
    //расчитываем точность азимута, необходимую для отображения метки
    private List<Double> calculateAzimuthAccuracy( double azimuth) {
        double minAngle = azimuth - AZIMUTH_ACCURACY;
        double maxAngle = azimuth + AZIMUTH_ACCURACY;
        List<Double> minMax = new ArrayList<Double>();

```

```

        if (minAngle < 0)
            minAngle += 360;

        if (maxAngle >= 360)
            maxAngle -= 360;

        minMax.clear();
        minMax.add(minAngle);
        minMax.add(maxAngle);

        return minMax;
    }
    //Метод isBetween определяет, находится ли азимут в целевом диапазоне с учетом
    допустимых отклонений

    private boolean isBetween( double minAngle, double maxAngle, double azimuth) {
        if (minAngle > maxAngle) {
            if (isBetween( 0, maxAngle, azimuth) && isBetween(minAngle, 360 , azi-
muth))
                return true ;
        } else {
            if (azimuth > minAngle && azimuth < maxAngle)
                return true ;
        }
        return false;
    }
    // выводим на экран основную информацию о местоположении цели и нашего
    устройства

    private void updateDescription() {

        long distance = ( long ) calculateDistance();
        int tAzimut = ( int ) mAzimuthTeoretical ;
        int rAzimut = ( int ) mAzimuthReal ;

        String text = mPoi.get KL9Name()
            + " location:"
            + "\n latitude: " + TARGET_LATITUDE + " longitude: " + TAR-
GET_LONGITUDE
            + "\n Current location:"
            + "\n Latitude: " + mMyLatitude + " Longitude: " + mMyLongi-
tude
            + "\n "
            + "\n Target azimuth: " + tAzimut
            + " \n Current azimuth: " + rAzimut
            + " \n Distance: " + distance;

        descriptionTextView.setText(text);
    }

```

/\*переопределяем метод слушателя OnAzimuthChangeListener, который вызывается при изменении азимута устройства, рассчитанного на основании показаний датчиков, получаемых в параметрах этого метода из класса MyCurrentAsimuth. Получаем данные азимута устройства, сравниваем их с целевыми параметрами - проверяем, если азимуты реальный и теоретический, а также дистанция до цели совпадают в пределах допустимых значений, отображаем метку на экране. Также вызываем метод обновления информации о местоположении на экране.\*/

```
@Override
public void onAzimuthChanged( float azimuthChangedFrom, float azimuthChangedTo) {
    mAzimuthReal = azimuthChangedTo;
    mAzimuthTeoretical = calculateTeoreticalAzimuth();
    int distance = ( int ) calculateDistance();

    pointerIcon = (ImageView) findViewById(R.id. icon );

    double minAngle = calculateAzimuthAccuracy(mAzimuthTeoretical ).get( 0);
    double maxAngle = calculateAzimuthAccuracy(mAzimuthTeoretical ).get( 1);

    if ((isBetween(minAngle, maxAngle, mAzimuthReal )) && distance <= DIS-
TANCE_ACCURACY ) {
        pointerIcon.setVisibility(View. VISIBLE );
    } else {
        pointerIcon.setVisibility(View. INVISIBLE );
    }

    updateDescription();
}
/*переопределяем метод onLocationChanged интерфейса слушателя OnLocation-
ChangedListener, здесь при изменении местоположения отображаем тост с новыми координатами и вызываем метод, который выводит основную информацию на экран.*/
```

```
@Override
public void onLocationChanged(Location location) {
    mMyLatitude = location.getLatitude();
    mMyLongitude = location.getLongitude();
    mAzimuthTeoretical = calculateTeoreticalAzimuth();
    Toast.makeText (this , "latitude: "+location.getLatitude()+ " longitude:
"+location.getLongitude(), Toast. LENGTH_SHORT ).show();
```

//если устройство возвращает азимут = 0 отображаем метку на основе значения дистанции

```
if (mAzimuthReal == 0){
    if ( distance <= DISTANCE_ACCURACY) {
        pointerIcon.setVisibility(View.VISIBLE);
    } else {
        pointerIcon.setVisibility(View.INVISIBLE);
    }
}
```

```

        }

        updateDescription();
    }
    /*в методе жизненного цикла onStop мы вызываем методы отмены регистрации дат-
чика азимута и закрытия подключения к службам Google Play*/

    @Override
    protected void onStop() {
        myCurrentAzimuth.stop();
        myCurrentLocation.stop();
        super.onStop();
    }
    //в методе onResume соответственно открываем подключение и регистрируем
слушатель датчиков

    @Override
    protected void onResume() {
        super.onResume();
        myCurrentAzimuth.start();
        myCurrentLocation.start();
    }
    /*метод setupListeners служит для инициализации слушателей местоположения и ази-
мута – здесь мы вызываем конструкторы классов MyCurrentLocation и MyCurrentAzimuth и
выполняем их методы start*/

    private void setupListeners() {
        myCurrentLocation = new MyCurrentLocation( this);
        myCurrentLocation.buildGoogleApiClient( this );
        myCurrentLocation.start();

        myCurrentAzimuth = new MyCurrentAzimuth( this, this);
        myCurrentAzimuth.start();
    }
    //метод setupLayout инициализирует все элементы экрана и создает surfaceView для
отображения превью камеры

    private void setupLayout() {
        descriptionTextView = (TextView) findViewById(R.id.cameraTextView );
        btnMap = (Button) findViewById(R.id. btnMap );
        btnMap.setVisibility(View. VISIBLE );
        btnMap.setOnClickListener( this );
        getWindow().setFormat(PixelFormat. UNKNOWN);
        SurfaceView surfaceView = (SurfaceView) findViewById(R.id.cameraview );
        mSurfaceHolder = surfaceView.getHolder();
        mSurfaceHolder.addCallback( this );
        mSurfaceHolder.setType(SurfaceHolder. SURFACE_TYPE_PUSH_BUFFERS );
    }
    /*вызывается сразу же после того, как были внесены любые структурные изменения
(формат или размер) surfaceView.*/

```



```

@Override
public void surfaceChanged(SurfaceHolder holder, int format, int width,
                           int height) {
    if ( isCameraviewOn ) {
        mCamera.stopPreview();
        isCameraviewOn = false ;
    }

    if ( mCamera != null ) {
        try {
            mCamera .setPreviewDisplay( mSurfaceHolder);
            mCamera .startPreview();
            isCameraviewOn = true ;
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
/*вызывается при первом создании surfaceView, здесь получаем доступ к камере и
устанавливаем ориентацию дисплея превью*/

@Override
public void surfaceCreated(SurfaceHolder holder) {
    mCamera = Camera. open();
    mCamera.setDisplayOrientation( 90);
}
//вызывается перед уничтожением surfaceView, останавливаем превью и освобождаем
камеру

@Override
public void surfaceDestroyed(SurfaceHolder holder) {
    mCamera.stopPreview();
    mCamera.release();
    mCamera = null ;
    isCameraviewOn = false ;
}
}

```

В методе onCreate вызываем методы setupListeners(); setupLayout(); setAugmentedRealityPoint().

Нам нужен доступ к камере, доступ к интернету и сетевым подключениям. Для определения местоположения прописаны такие строчки:

android.permission.ACCESS\_FINE\_LOCATION – позволяет максимально точно определять местоположение, используя все доступные способы: GPS, Wi-Fi и сеть сотовой связи

ACCESS\_COARSE\_LOCATION — позволяет приложению получить доступ к приблизительному местоположению.

com.google.android.providers.gsf.permission.READ\_GSERVICES позволяет работать с гуголкартами.

Также здесь видим секцию meta-data, где прописаны версия google\_play\_services и API\_KEY для работы Google карт.

## 2.2 Описание продукта (результата разработки)

Приложение готово к запуску. Тестировать его нужно на реальном устройстве. Запускаем приложение и видим на экране интерфейс представленные на рисунке 7. После выбора учебного корпуса открывается камера (смотри

Рисунок 8. Интерфейс программы «Путеводитель УрГПУ»). При наведении камеры на метку на ней отображается информация о местоположении и расстоянии до нее. Примеры представлены на Рисунок 11, Рисунок 12, Рисунок 13

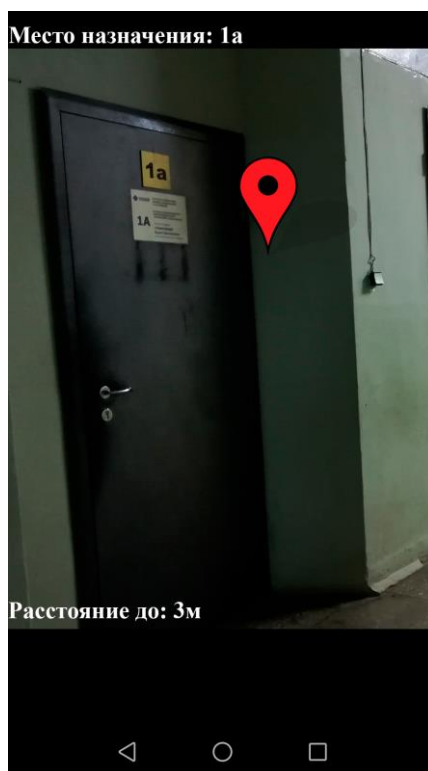


Рисунок 11. Интерфейс программы «Путеводитель УрГПУ»



Рисунок 12. Интерфейс программы «Путеводитель УрГПУ»



Рисунок 13. Интерфейс программы «Путеводитель УрГПУ»

Дистанция отображается в метрах. В своем приложении я указала для кабинетов местоположение, которое совпадает с их расположением. При совпа-

дении азимутов в пределах константы в поле зрения камеры на экране появляется изображение метки. Если азимуты не совпадают в пределах константы в поле зрения камеры возникает стрелка по минимальному углу до метки.

## 2.3 Результаты апробации, техническая документация

Приложение распространялось по ссылке пользователям, смартфоны которых удовлетворяли техническим параметрам. После тестирования программы пользователям предлагалось пройти электронный опрос. На рисунке 14 представлена диаграмма общей оценки программы.

### Оцените работу программы

9 ответов

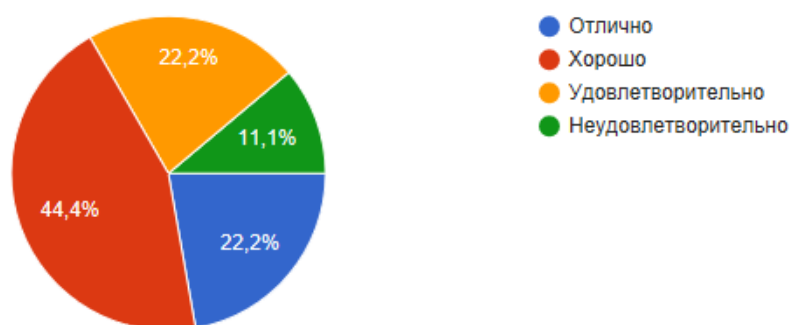


Рисунок 14. Диаграмма общей оценки программы

Оценка интерфейса программы представлена на рисунке 15.

### Оцените интерфейс программы

9 ответов

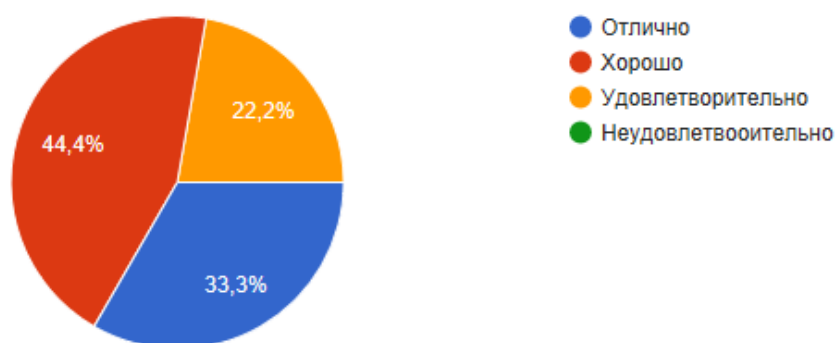


Рисунок 15. Диаграмма оценки интерфейса программы

Соответствие программы ее назначению представлено на рисунке 16. Видно, что программа выполняет функции, которые возлагались на нее по техническому заданию.

#### Оцените соответствие функционала и назначения

9 ответов

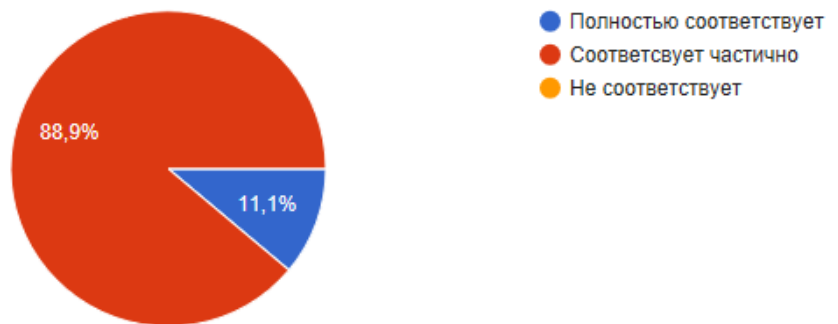


Рисунок 16. Диаграмма оценки соответствия функционала и назначения

Удобство программы также было оценено пользователями. Результаты представлены на рисунке 17.

#### Удобна ли программа для пользования

9 ответов

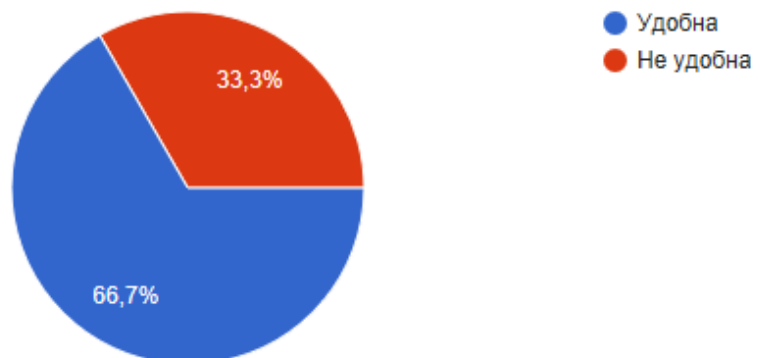


Рисунок 17. Диаграмма оценки удобства программы

Участники тестирования высказались об общей полезности программного продукта, однако, с ней необходимо провести доработки. Результаты оценки полезности представлены на Рисунок 18.

### Полезна ли программа

9 ответов

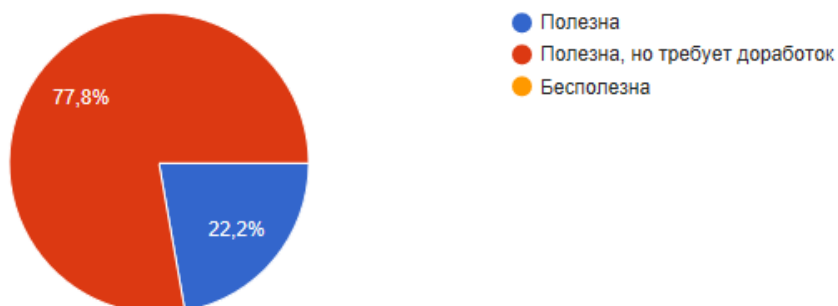


Рисунок 18. Диаграмма оценки полезности

Так же было выяснено, что на многих устройствах возникали ошибки и сбои в работе программы. Результаты опроса по ошибкам представлены на рисунке 19.

### Появлялись ли ошибки при работе программы

9 ответов

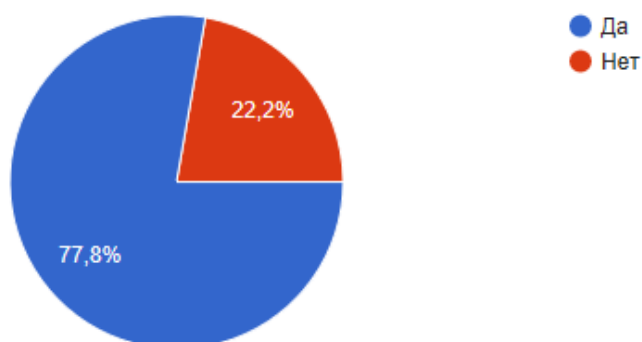


Рисунок 19. Диаграмма оценки ошибок

Вывод, который можно сделать по данному: программа имеет набор функций, закладываемый в нее техническим заданием, однако, она требует доработки и оптимизации под большее число устройств. Так же было выяснено, что неудобство интерфейса было вызвано несоответствием разрешения экрана на устройствах тестеров и разработчика.

## **Заключение**

В результате исследования было показано что программы с использованием технологии дополненной реальности являются актуальными в широком спектре областей (от военных технологий до игровых приложений), так же важным направлением является навигационная область. Программа «путеводитель УрГПУ» была создана в ходе выполнения данной работы. Таким образом цель была достигнута.

В ходе выполнения работы:

- 1) Проведен анализ существующих программных продуктов.
- 2) Был выбран способ реализации собственного программного продукта.
- 3) Разработана программа «путеводитель УрГПУ»
- 4) Составлена пояснительная записка по данной работе.

На основе статистических данных была произведена оценка продукта у ограниченного числа пользователей из которой видно, что данный программный продукт соответствует техническому заданию.

## **Список информационных источников**

1. Оформитель библиографических ссылок // SNOSKA.INFO URL: <http://snoskainfo.ru/> (дата обращения: 14.02.2019).
2. ГОСТ Р 7.0.83-2013 «Система стандартов по информации, библиотечному и издательскому делу. Электронные издания. Основные виды и выходные сведения» от 15.10.2013 № 1163-ст // Стандартиформ. 2014 г. с изм. и доп. в ред. от 12.09.2018.
3. ГОСТ 2.105-95. Межгосударственный стандарт «Единая система конструкторской документации. Общие требования к текстовым документам» от 08.08.1995 № 426 // Всероссийским научно-исследовательским институтом стандартизации и сертификации в машиностроении (ВНИИНМАШ) Госстандарта России. 1995 г.
4. ГОСТ 19.502-78. государственный стандарт Союза ССР «Единая система программной документации. Описание применения. Требования к содержанию и оформлению.» от 18.12.1978 № 3350 // Издательство стандартов. с изм. и доп. в ред. от 01.01.1982.
5. ГОСТ 19.503-79 «Единая система программной документации. Руководство системного программиста. Требования к содержанию и оформлению» от 12.01.1979 № 74 // Издательство стандартов. с изм. и доп. в ред. от 01.01.1982.
6. ГОСТ 19.504-79 «Единая система программной документации. Руководство программиста. Требования к содержанию и оформлению» от 12.01.1979 № 74 // Издательство стандартов. с изм. и доп. в ред. от 01.01.1982.
7. ГОСТ 19.505-79 «Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению.» от 12.01.1979 № 74 // Издательство стандартов. с изм. и доп. в ред. от 01.01.1982.
8. ГОСТ Р 53620-2009 «Информационно-коммуникационные технологии в образовании. Электронные образовательные ресурсы. Общие положения»



- от 15.12.2009 № 956-ст // Стандартиформ. 2011 г. с изм. и допол. в ред. от 10.12.2018.
9. ГОСТ Р 52653-2006 «Информационно-коммуникационные технологии в образовании. Термины и определения» от 27.12.2006 № 419-ст // Стандартиформ. 2007 г. с изм. и допол. в ред. от 10.12.2018.
  10. ГОСТ Р 52657-2006 «Информационно-коммуникационные технологии в образовании. Образовательные интернет-порталы федерального уровня. Рубрикация информационных ресурсов» от 27.12.2006 № 423-ст // Стандартиформ. 2007 г. с изм. и допол. в ред. от 12.09.2018.
  11. Санитарно-эпидемиологические правила и нормативы СанПиН 2.4.2.2821-10 «Санитарно-эпидемиологические требования к условиям и организации обучения в общеобразовательных организациях» от 29.12.2010 № 189 // Роспотребнадзор. с изм. и допол. в ред. от 24.11.2015.
  12. Дополненная реальность (AR): перспективы и будущее технологии // Комсомольская правда URL:  
<https://www.kp.ru/putevoditel/tekhnologii/dopolnennaya-realnost/> (дата обращения: 1.05.2019).
  13. Дополненная реальность // Википедия URL:  
[https://ru.wikipedia.org/wiki/Дополненная\\_реальность](https://ru.wikipedia.org/wiki/Дополненная_реальность) (дата обращения: 1.05.2019).
  14. Мытников Александр Николаевич, Мытникова Екатерина Анатольевна  
ИСТОРИЯ РАЗВИТИЯ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ // НоваУм. 2017. №5.
  15. Увлекательная реальность URL:  
[http://funreality.ru/technology/augmented\\_reality/](http://funreality.ru/technology/augmented_reality/) (дата обращения: 02.05.2019).
  16. Захват движения // Википедия URL:  
[https://ru.wikipedia.org/wiki/Захват\\_движения](https://ru.wikipedia.org/wiki/Захват_движения) (дата обращения: 02.05.2019).

17. Кравцов А.А ИССЛЕДОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ С ТЕХНОЛОГИЕЙ ИНТЕРАКТИВНОЙ ВИЗУАЛИЗАЦИИ СРЕДСТВАМИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ: дис. ... канд. тех. наук: 05.13.01. Краснодар, 2016.
18. Карина Билялутдинова Семь основных методологий разработки // Старт высоких достижений. 2017. №5.
19. Марио Цехнер Программирование игр под Android. СПб: ООО Издательство "Питер", 2013.
20. ТЕХНОЛОГИЯ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ КАК СРЕДСТВО ПОДДЕРЖКИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА // Национальный исследовательский технологический университет "МИСиС" URL: <http://dgng.pstu.ru/conf2017/papers/110/> (дата обращения: 10.05.2019).